

1. Kudu核心API

1.1. 客户端API核心类

- 1.1.1. [Kudu client](#)
- 1.1.2. [Schema](#)
- 1.1.3. [ColumnSchema](#)
- 1.1.4. [CreateTableOptions](#)
- 1.1.5. [KuduTable](#)
- 1.1.6. [Session](#)
- 1.1.7. [Insert/Update/Delete/Upsert](#)
- 1.1.8. [PartialRow](#)
- 1.1.9. [KuduScanner](#)

2. Java编程接口

2.1. 创建maven工程，加入kudu依赖

2.2. 基本操作

- 2.2.1. [创建表](#)
- 2.2.2. [插入数据源](#)
- 2.2.3. [查询数据](#)
- 2.2.4. [修改表结构](#)
- 2.2.5. [更新数据](#)
- 2.2.6. [删除数据](#)
- 2.2.7. [更新和插入](#)
- 2.2.8. [删除表](#)

2.3. 其他语言编程接口

1. Kudu核心API

这里介绍 Kudu 提供的主流语言的客户端 API，它主要用于构建上层 **非分析类应用**。

1.1. 客户端API核心类

Kudu 提供了主流语言客户端API，核心类、方法是一致的，我们首先简要盘点下核心的这些类以便于我们写代码。

1.1.1. Kudu client

AsyncKuduClient：完全异步且线程安全的 Kudu 客户端。该类应该只实例化一次，同时访问很多表。只有操纵多个不同集群才需要实例化多次。不会阻塞操作，可以关联回调函数用于操作完成时的动作。Builder 模式创建。

KuduClient：对 AsyncKuduClient 的封装，同步执行、线程安全的 Kudu 客户端，Builder 模式创建。

1.1.2. Schema

表示表结构，主要是 column 的集合。该类提供了一些工具方法用于查询操作。

1.1.3. ColumnSchema

表示一个列的定义 column，使用 builder 模式 构建。

1.1.4. CreateTableOptions

Builder 模式类，用于创建表。

1.1.5. KuduTable

表示集群上的一张表。含有当前表结构信息，隶属于特定 AsyncKuduClient。

1.1.6. Session

AsyncKuduSession：隶属于特定 KuduClient，代表一个上下文环境，所有写操作都会在该上下文中进行。在一个 session 中，可以对多个操作按批处理方式执行，以获得较好的性能。每个 session 都可以设置超时、优先级以及跟踪 id 等信息。session 和 KuduClient 是独立的，主要是在多线程环境下，不同线程需要并发执行事务，事务的边界是基于每个 session 的 BeginTransaction 和 commit 之间的过程。来自于不同 session 的写操作不会组织到一个 RPC 请求 batch 中，意味着延迟敏感的客户端（低延迟）和面向吞吐量的客户端（高延迟）使用同一 KuduClient，每个 Session 中可以设置特定的超时和优先级。

KuduSession：对 AsyncKuduSession 封装，同步执行，但非线程安全。

1.1.7. Insert/Update/Delete/Upsert

表示插入/更新/删除/插入或者更新操作，对象不可复用。

1.1.8. PartialRow

表示一行的部分列。

1.1.9. KuduScanner

扫描对象，用于条件查询及迭代获取结果集。

2. Java编程接口

2.1. 创建maven工程，加入kudu依赖

```
<properties>
  <project.build.sourceEncoding>UTF-8</project.build.sourceEncoding>
  <maven.compiler.source>1.8</maven.compiler.source>
  <maven.compiler.target>1.8</maven.compiler.target>
  <kudu.version>1.11.1</kudu.version>
  <junit.version>4.13</junit.version>
</properties>

<dependencies>
  <dependency>
    <groupId>junit</groupId>
    <artifactId>junit</artifactId>
    <version>${junit.version}</version>
  </dependency>

  <!-- Kudu client -->
  <dependency>
    <groupId>org.apache.kudu</groupId>
    <artifactId>kudu-client</artifactId>
    <version>${kudu.version}</version>
  </dependency>

  <!-- Log -->
  <dependency>
    <groupId>org.slf4j</groupId>
    <artifactId>slf4j-simple</artifactId>
    <version>1.7.12</version>
  </dependency>
</dependencies>

<!-- 指定具体仓库 -->
<repositories>
  <repository>
    <id>cdh.repo</id>
    <name>Cloudera Repositories</name>
    <url>https://repository.cloudera.com/content/repositories/releases</url>
    <snapshots>
      <enabled>>false</enabled>
    </snapshots>
  </repository>
</repositories>
```

2.2. 基本操作

定义一个常量 KUDU_MASTERS 存放 Kudu master 的连接信息(根据自己的实际情况改为你的IP或者主机名):

```
private static final String KUDU_MASTERS =
    "bigdata02:7051,bigdata03:7051,bigdata04:7051";
```

2.2.1. 创建表

```
package com.mazh.kudu.core.crud;

import com.mazh.kudu.constant.Kudu_Constant;
import org.apache.kudu.ColumnSchema;
import org.apache.kudu.Schema;
import org.apache.kudu.Type;
import org.apache.kudu.client.CreateTableOptions;
import org.apache.kudu.client.KuduClient;
import org.apache.kudu.client.KuduException;

import java.util.ArrayList;
import java.util.List;

/**
 * Author: 马中华 奈学教育 https://blog.csdn.net/zhongqi2513
 * DateTime: 2020/06/11 14:48
 * Description:
 */
public class Kudu_App01_CreateTable {

    public static void main(String[] args) {
        // 请开始你的表演!

        /**
         * TODO 第一步: 创建Schema
         */
        List<ColumnSchema> columns = new ArrayList<ColumnSchema>(3);
        columns.add(new ColumnSchema.ColumnSchemaBuilder("uid",
            Type.INT8).key(true).build());
        columns.add(new ColumnSchema.ColumnSchemaBuilder("name",
            Type.STRING).nullable(true).build());
        columns.add(new ColumnSchema.ColumnSchemaBuilder("age",
            Type.INT8).build());
        Schema schema = new Schema(columns);

        /**
         * TODO 第二步: 指定表选项
         */
        CreateTableOptions tableOptions = new CreateTableOptions();

        /**
         * TODO 第三步: 创建分区策略
         */
        List<String> hashColumns = new ArrayList<String>();
        hashColumns.add("uid");
        int numBuckets = 6;
        tableOptions.addHashPartitions(hashColumns,
            numBuckets).setNumReplicas(1);

        /**
         * TODO 第四步: 创建 KuduClient
         */
    }
}
```

```

        KuduClient client = null;

        try {
            client = new
KuduClient.KuduClientBuilder(Kudu_Constant.KUDU_MASTERS).build();
            if(!client.tableExists("users")) {
                client.createTable("users", schema, tableOptions);
                System.out.println(".....create table success.....");
            } else {
                System.out.println(".....the table already exists
.....");
            }
        } catch(KuduException e) {
            e.printStackTrace();
        } finally {
            /**
             * TODO 第五步: 关闭资源
             */
            if(null != client) {
                try {
                    client.shutdown();
                } catch(KuduException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}

```

2.2.2. 插入数据源

```

package com.mazh.kudu.core.crud;

import org.apache.kudu.client.*;

import static com.mazh.kudu.constant.Kudu_Constant.KUDU_MASTERS;

/**
 * Author: 马中华 奈学教育 https://blog.csdn.net/zhongqi2513
 * DateTime: 2020/06/11 14:57
 * Description:
 */
public class Kudu_App02_InsertData {

    public static void main(String[] args) {
        // 请开始你的表演!
        // 1、获得kudu客户端
        KuduClient client = null;

        // 2、打开表
        KuduTable table = null;

        // 3、创建会话
        KuduSession session = null;
        try {

```

```

        client = new KuduClient.KuduClientBuilder(KUDU_MASTERS).build();
        table = client.openTable("users");
        session = client.newSession();

        session.setFlushMode(SessionConfiguration.FlushMode.AUTO_FLUSH_BACKGROUND);
        session.setFlushInterval(2000);

        // 4、循环插入10行记录
        for(int i = 0; i < 100; i++) {
            //新建Insert对象
            Insert insert = table.newInsert();
            PartialRow row = insert.getRow();
            row.addByte("uid", Byte.parseByte(i + ""));
            //i是偶数
            if(i % 2 == 0) {
                row.setNull("name");
            } else {
                row.addString("name", "name " + i);
            }
            row.addByte("age", Byte.parseByte(i + ""));
        }

        // 5、关闭session session.close() ;
        //判断错误数if(session.countPendingErrors() != 0){
        //获得操作结果
        RowErrorsAndOverflowStatus result = session.getPendingErrors();
        if(result.isOverflowed()) {
            System.out.println(".....buffer溢出!.....");
        }
        RowError[] errs = result.getRowErrors();
        for(RowError er : errs) {
            System.out.println(er);
        }
    } catch(KuduException e) {
        e.printStackTrace();
    } finally {
        if(null != client) {
            try {
                client.shutdown();
            } catch(KuduException e) {
                e.printStackTrace();
            }
        }
    }
}
}
}

```

2.2.3. 查询数据

```

package com.mazh.kudu.core.crud;

import org.apache.kudu.Schema;
import org.apache.kudu.client.*;

import java.util.ArrayList;

```

```

import java.util.List;

import static com.mazh.kudu.constant.Kudu_Constant.KUDU_MASTERS;

/**
 * Author: 马中华 奈学教育 https://blog.csdn.net/zhongqi2513
 * DateTime: 2020/06/11 15:03
 * Description:
 */
public class Kudu_App03_Select {

    public static void main(String[] args) {
        // 请开始你的表演!
        //1、获得kudu客户端
        KuduClient client = new
KuduClient.KuduClientBuilder(KUDU_MASTERS).build();

        try {
            //2、打开表
            kuduTable table = client.openTable("users");

            //3、扫描器
            kuduScanner scanner = null;

            //4、获取表结构
            Schema schema = table.getSchema();

            //5、指定查询条件
            List<String> projectColumns = new ArrayList<String>(2);
            projectColumns.add("uid");
            projectColumns.add("name");
            projectColumns.add("age");

            //age >= 0
            int lowerBound = 0;
            KuduPredicate lowerPred = KuduPredicate
                .newComparisonPredicate(schema.getColumn("age"),
KuduPredicate.ComparisonOp.GREATER_EQUAL, lowerBound);

            //age < 10
            int upperBound = 10;
            KuduPredicate upperPred =
KuduPredicate.newComparisonPredicate(schema.getColumn("age"),
KuduPredicate.ComparisonOp.LESS, upperBound);

            scanner =
client.newScannerBuilder(table).setProjectedColumnNames(projectColumns).addPredi
cate(lowerPred).addPredicate(upperPred).build();

            int resultCount = 0;
            while(scanner.hasMoreRows()) {
                RowResultIterator results = scanner.nextRows();
                while(results.hasNext()) {
                    RowResult result = results.next();
                    byte uid = result.getByte("uid");
                    String name = null;
                    if(result.isNull("name")) {
                        name = "不存在";
                    }
                }
            }
        }
    }
}

```

```

        } else {
            name = result.getString("name");
        }
        byte age = result.getByte("age");
        System.out.printf("uid=%d, name=%s, age=%d\r\n", uid, name,
age);

        resultCount++;
    }
}

System.out.println("-----" + resultCount);
scanner.close();

} catch(KuduException e) {
    e.printStackTrace();
} finally {
    if(null != client) {
        try {
            client.shutdown();
        } catch(KuduException e) {
            e.printStackTrace();
        }
    }
}
}
}
}

```

2.2.4. 修改表结构

```

package com.mazh.kudu.core.crud;

import org.apache.kudu.Type;
import org.apache.kudu.client.AlterTableOptions;
import org.apache.kudu.client.KuduClient;
import org.apache.kudu.client.KuduException;

import static com.mazh.kudu.constant.Kudu_Constant.KUDU_MASTERS;

/**
 * Author: 马中华 奈学教育 https://blog.csdn.net/zhongqi2513
 * DateTime: 2020/06/11 15:05
 * Description:
 */
public class Kudu_App04_Alter {

    public static void main(String[] args) {
        // 请开始你的表演!
        //1、获得kudu客户端
        KuduClient client = null;

        try {
            //2、修改表选项
            AlterTableOptions ato = new AlterTableOptions();
            ato.addColumn("wage", Type.DOUBLE, 10000.000);

```



```

        //3、修改表结构
        client = new KuduClient.KuduClientBuilder(KUDU_MASTERS).build();
        if(client.tableExists("users")) {
            client.alterTable("users", ato);
            System.out.println(".....alterTable success.....");
        }
    } catch(KuduException e) {
        e.printStackTrace();
    } finally {
        //4、关闭资源if(null!=client){
        try {
            client.shutdown();
        } catch(KuduException e) {
            e.printStackTrace();
        }
    }
}
}
}

```

修改完再次查询:

```

@Test
public void Kudu_App04_Select() throws KuduException {
    //1、获得kudu客户端
    KuduClient client = new KuduClient.KuduClientBuilder(KUDU_MASTERS).build();

    //2、打开表
    KuduTable table = client.openTable("users") ;

    //3、扫描器
    KuduScanner scanner = null; try {

        //4、获取表结构
        Schema schema = table.getSchema() ;

        //5、指定查询条件
        List<String> projectColumns = new ArrayList<String>(2);
        projectColumns.add("uid");
        projectColumns.add("name");
        projectColumns.add("age");
        projectColumns.add("wage");

        //age >= 0
        int lowerBound = 0;
        KuduPredicate lowerPred =
            KuduPredicate.newComparisonPredicate(schema.getColumn("age"),
            KuduPredicate.ComparisonOp.GREATER_EQUAL, lowerBound);

        //age < 10
        int upperBound = 10;
        KuduPredicate upperPred =
            KuduPredicate.newComparisonPredicate(schema.getColumn("age"),
            KuduPredicate.ComparisonOp.LESS, upperBound);

        scanner = client.newScannerBuilder(table)
            .setProjectedColumnNames(projectColumns)
            .addPredicate(lowerPred)

```

```

        .addPredicate(upperPred)
        .build();

    int resultCount = 0;
    while (scanner.hasMoreRows()) {
        RowResultIterator results = scanner.nextRows();
        while (results.hasNext()) {
            RowResult result = results.next();
            byte uid = result.getByte("uid");
            String name = null ;
            if (result.isNull("name")) {
                name = "不存在" ;
            }
            else{
                name = result.getString("name") ;
            }
            byte age = result.getByte("age");
            double wage = result.getDouble("wage");
            System.out.printf("uid=%d, name=%s, age=%d, wage=%f\r\n" ,
uid,name,age,wage);
            resultCount++;
        }
    }
    System.out.println("-----"+resultCount);
    scanner.close() ;
}finally {
    if(null!=client){
        client.shutdown();
    }
}
}
}

```

2.2.5. 更新数据

```

package com.mazh.kudu.core.crud;

import org.apache.kudu.client.*;

import static com.mazh.kudu.constant.Kudu_Constant.KUDU_MASTERS;

/**
 * Author: 马中华 奈学教育 https://blog.csdn.net/zhongqi2513
 * DateTime: 2020/06/11 15:08
 * Description:
 */
public class Kudu_App04_Update {

    public static void main(String[] args) {
        // 请开始你的表演!
        //1、获得kudu客户端
        KuduClient client = null;
        //2、打开表
        KuduTable table = null;
    }
}

```

```

//3、会话
KuduSession session = null;
try {
    client = new KuduClient.KuduClientBuilder(KUDU_MASTERS).build();
    table = client.openTable("users");
    session = client.newSession();
    //4、创建并执行update操作
    Update update = table.newUpdate();
    PartialRow row = update.getRow();
    row.addByte("uid", Byte.parseByte("1" + ""));
    row.addDouble("wage", 20000.000);
    session.apply(update);
    session.close();
} catch (KuduException e) {
    e.printStackTrace();
} finally {
    //5、关闭资源if(null!=client){
    try {
        client.shutdown();
    } catch (KuduException e) {
        e.printStackTrace();
    }
}
}
}

```

2.2.6. 删除数据

```

package com.mazh.kudu.core.crud;

import org.apache.kudu.client.*;

import static com.mazh.kudu.constant.Kudu_Constant.KUDU_MASTERS;

/**
 * Author: 马中华 奈学教育 https://blog.csdn.net/zhongqi2513
 * DateTime: 2020/06/11 15:09
 * Description:
 */
public class Kudu_App05_Delete {

    public static void main(String[] args) {
        // 请开始你的表演!
        //1、获得kudu客户端
        KuduClient client = null;

        //2、打开表
        KuduTable table = null;

        //3、会话
        KuduSession session = null;
        try {
            client = new KuduClient.KuduClientBuilder(KUDU_MASTERS).build();
            table = client.openTable("users");
            session = client.newSession();

```

```

        //4、新建并执行Delete操作
        Delete delete = table.newDelete();
        //得到row
        PartialRow row = delete.getRow();

        //where key = 0
        row.addByte("uid", Byte.parseByte(3 + ""));
        session.apply(delete);

        session.close();
    } catch (KuduException e) {
        e.printStackTrace();
    } finally {
        //5、关闭资源if(null!=client){
        try {
            client.shutdown();
        } catch (KuduException e) {
            e.printStackTrace();
        }
    }
}
}
}

```

2.2.7. 更新和插入

```

package com.mazh.kudu.core.crud;

import org.apache.kudu.client.*;

import static com.mazh.kudu.constant.Kudu_Constant.KUDU_MASTERS;

/**
 * Author: 马中华 奈学教育 https://blog.csdn.net/zhongqi2513
 * DateTime: 2020/06/11 15:10
 * Description:
 */
public class Kudu_App06_Upsert {

    public static void main(String[] args) {
        // 请开始你的表演!
        //1、获得kudu客户端
        KuduClient client = null;

        //2、打开表
        KuduTable table = null;

        //3、会话
        KuduSession session = null;
        try {
            client = new KuduClient.KuduClientBuilder(KUDU_MASTERS).build();
            table = client.openTable("users");
            session = client.newSession();

            //4、upsert

```

```

        Upsert upsert = table.newUpsert();
        PartialRow row = upsert.getRow();
        row.addByte("uid", Byte.parseByte(3 + ""));
        row.addString("name", "tomasLee");
        row.addByte("age", Byte.parseByte(35 + ""));
        row.addDouble("wage", 18000.000);
        session.apply(upsert);

        Upsert upsert1 = table.newUpsert();
        PartialRow row1 = upsert1.getRow();
        row1.addByte("uid", Byte.parseByte(1 + ""));
        row1.addByte("age", Byte.parseByte(8 + ""));
        row1.addDouble("wage", 15000.000);
        session.apply(upsert1);

        session.close();
    } catch (KuduException e) {
        e.printStackTrace();
    } finally {
        //5、关闭资源if(null!=client){
        try {
            client.shutdown();
        } catch (KuduException e) {
            e.printStackTrace();
        }
    }
}
}

```

注意：Upsert主键一样则更新，否则为新增，不能为空的字段必须提供值，否则不执行。

2.2.8. 删除表

```

package com.mazh.kudu.core.crud;

import org.apache.kudu.client.KuduClient;
import org.apache.kudu.client.KuduException;

import static com.mazh.kudu.constant.Kudu_Constant.KUDU_MASTERS;

/**
 * Author: 马中华 奈学教育 https://blog.csdn.net/zhongqi2513
 * DateTime: 2020/06/11 15:12
 * Description:
 */
public class Kudu_App07_DropTable {

    public static void main(String[] args) {
        // 请开始你的表演!
        // 1、创建 KuduClient
        KuduClient client = null;

        try {
            client = new KuduClient.KuduClientBuilder(KUDU_MASTERS).build();

```

```
        //2、删除表if(client.tableExists("users")){
            client.deleteTable("users");
            System.out.println(".....delete table success.....");
        } catch(KuduException e) {
            e.printStackTrace();
        } finally {
            //3、关闭资源if(null!=client){
                try {
                    client.shutdown();
                } catch(KuduException e) {
                    e.printStackTrace();
                }
            }
        }
    }
}
```

2.3. 其他语言编程接口

请参看官方examples: <https://github.com/apache/kudu/tree/master/examples>